

# 通訊伺服器之驅動程式製作

## The Driver Implementation for Communication Server

黃圳柏 黃文祥 羅啟文

國立高雄應用科技大學電機系

E-mail : adrian@wshlab2.ee.kuas.edu.tw, wshwang@mail.ee.kuas.edu.tw

ken@wshlab2.ee.kuas.edu.tw

### 摘要

通訊應用範疇的課題已進展到了多元化，舉凡設備、家電、移動式工作平台，均朝向整合式通訊系統方向發展。本文將敘述一個具多埠通訊伺服器在 Linux 下開發其此驅動程式及存取上層協定如 PPP, SNMP等；並建立 Web Server 及 IP 分享之實作。文中將著重於驅動程式部份的探討，並對實作結果做詳細描述與討論。

關鍵詞：通訊伺服器、驅動程式、IP 分享

### 1. 前言

由於通訊科技已深入各個階層，讓各類的非通訊設備、家用電器、或可移動式的工作平台經由網路連結而相互通訊，進而達到遠端遙控的目的；其中的技術包含有無線傳輸(如 IEEE 802.11b、藍芽技術、紅外線傳輸等)及有線傳輸(如 Ethernet、USB、RS-232、IEEE 488...等不勝枚舉)。然而既有的舊設備中除了 RS-232 序列埠外，少有配備上述的通訊裝置。因此如何將這類為數可觀的設備經由網際網路做有效率的遠端控制，是一個值得探究的課題。另外；由於這些設備須經由網際網路來傳輸，因此 IP 位址不足的問題也將發生在這個研討的課題中，解決這個問題的方法以 IPv6 及 NAT (Network Address Translation)為主，在近期的實作多以 NAT 為主。本文即針對這些課題提出以工業電腦在 Linux 平台上實現一台具 16 個序列埠的通訊伺服器並將於未來提供網頁遠端遙控及智慧型處理功能。使其可依即時性的資料擷取，經資料庫中預設的智識庫(Knowledge Base)做出第一時間

的處理，再通知負責人員做更進一步的處置。本文將敘述第一階段與全濠公司共同研究的部份，作者以製作該通訊伺服器之驅動程式，以達到驅動多埠序列卡之送收及上層協定之存取，進而達到 IP 分享之功能。

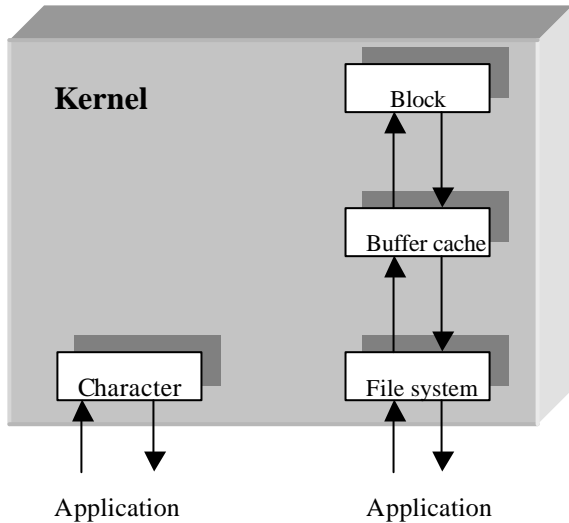
### 2. 背景

首先探討 linux 平台的裝置驅動程式(Device Driver)的製作原理。

#### 2-1 裝置分類

裝置驅動程式介於核心程式(Kernel)與其所控制硬體裝置間的一層介面(Interface)，其功能在於簡化 Kernel 對所有硬體裝置的管理工作，藉由提供 Kernel 一個完整通用介面，可將這些工作留給各個裝置驅動程式來個別處理，因此 Kernel 不需與每個裝置直接溝通，只要透過固定的驅動程式指令(Primitive instruction)即可與所欲控制硬體溝通。

裝置驅動程式依據其行為可分為兩種不同的裝置，一為字元式裝置(Character device)，字元式裝置係按位元組順序來進行存取，且無法以緩衝器系統(Buffer system)快取(Cache)，如同錄音帶中的聲音以連續資料流形式被讀取與撥放一般；這就是字元式裝置處理資料的方式。另一種為區塊式裝置(Block device)，這裝置所讀取的資料是以區塊為單位，並可透過緩衝快取系統進行存取。在大多數的 UNIX 系統上，一個區塊通常代表約一塊 1KB 資料；但基於效率考量，作業系統基本存取單位都包含好幾個區塊。圖一顯示字元式裝置與區塊式裝置於 Linux Kernel 中的差異性。



圖一、字元式裝置與區塊式裝置之差異性



圖二、通訊伺服器驅動卡之正視圖



圖三、通訊伺服器驅動卡之俯視圖

## 2-2 User mode 與 Kernel mode

程序(Process)在 Linux 系統下有兩種運作模式，即使用者模式(User mode)與系統管理者模式(supervisor mode)，後者又稱為核心模式(Kernel mode)，處理器是以這兩種模式來管理程序中指令的執行。一般程序運作的模式會受到作業系統對其使用的限制，例如使用者模式下之程序不能直接存取硬體或執行逾越其記憶體範圍的指令。大多數作業系統將各程式的運作空間分開，屬於作業系統運作的系統程式與一般使用者所執行之應用程式，各自佔有不同記憶空間，彼此不會互相干擾。裝置驅動程式或稱為模組(Modular)是在核心空間(Kernel space)上運作，應用程式則是在使用者空間(user space)上運作。

## 3. 通訊伺服器之硬體

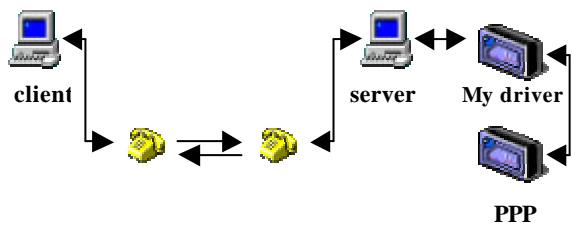
圖二、圖三為通訊伺服器驅動卡(ISA 插槽)之正視圖與俯視圖，此驅動卡以序列埠做為傳輸介面，本文主要探究撰寫這一張介面卡之驅動程式。圖四為整個通訊伺服器之架構圖，此通訊伺服器目前提供 16 ports 給使用者撥接至此通訊伺服器，當然還可視使用者所需增加至 32 ports、64 ports 等等。



圖四、通訊伺服器

## 4. 系統架構

系統架構如下頁圖五所示，當 Client 端透過電話線路撥接至 Server 端時，Client 端與 Server 端會先進行交握(Handshaking)程序，一旦交握成功之後，PPP driver 將被載入至系統，然後 PPP driver 會送 PPP 封包至通訊伺服器緩衝器裡，最後通訊伺服器再將這些資料經由 RS-232 線路(如圖四所示)傳送至 Client 端。當然 PPP driver 也會讀取從 Client 端送至 Server 端之 PPP 封包，並將這些 PPP 封包解析後，回應給 Client 端。同時 Client 端之 PPP driver 也會被載入至系統中，負責送出與讀取封包，並將之解析回應至對方，如此 Client 端就可以透過電話線路撥接至 Client 端，並連上 Internet，進而可達到此通訊伺服器具備 IP 分享之功能。



圖五、系統架構

## 5. 實作結果

目前已完成通訊伺服器介面卡之驅動程式撰寫，並經由各項實測以檢驗通訊伺服器介面卡的驅動以及資料的送收。目前通訊伺服器已具備兩項功能：1.通訊伺服器可與 minicom mgetty 等軟體溝通成功。2.可透過 minicom撥號至另一台 server，並在交握成功之後，遠端登入此 server。底下將說明通訊伺服器與 minicom 軟體溝通各個細節及如何遠端登入另一台 server。minicom起始畫面如圖 6 所示，圖中圈起來部份代表 minicom對數據機初始化的命令，當數據機接收到這些命令後，數據機會回應”OK”字串給 minicom軟體，代表此次所執行的命令已被成功地接收到。但必須注意 modem 在未建立連線時是處於 command mode，也就是送給 modem 的 data 會被當成 command 處理。當 modem 與遠方建立連線後，modem 就會進入 data mode，此時送給 modem 的字串都會被當成 data 送往遠端的 modem。

```

Welcome to minicom 1.83.1
OPTIONS: History Buffer, P-key Macros, Search History Buffer, 118s
Compiled on Oct  3 2000, 08:44:30.

Press CTRL-A & for help on special keys

AT E1=45 S0=0 L1 V1 B4 &cl B1 00
OK
  
```

圖六、minicom 起始畫面

接下來試著撥號至另一台 server，如圖七所示。圖七圈起來的地方可細分成兩個部分，atdt 指示數據機撥號出去，5566 則代表所欲撥的電話號碼。當 server 端接收到有人撥電話進來時，server 的 modem 會將之接起來，此時雙方 modem 正在進行交握動作(modem 會發出吵雜的聲音)，一旦交握成功之後，會出現如圖八所示。代表已成功地遠端登入本機。

```

Welcome to minicom 1.83.1
OPTIONS: History Buffer, P-key Macros, Search History Buffer, 118s
Compiled on Oct  3 2000, 08:44:30.

Press CTRL-A & for help on special keys

AT E1=45 S0=0 L1 V1 B4 &cl B1 00
OK
atdt 5566
  
```

圖七、撥號誌入另一台 server

```

Linux Version 2.4.8-26mdk
Compiled #1 Sat Sep 23 17:06: 17 EST 2001
One 750MHz AMD Duron Processor, 121M RAM
1507.32 Byteswap Total

adrian

Mandrake Linux release 8.1 (Villain) for i586
Kernel 2.4.8-26mdk on an i686 / i

adrian login:
  
```

圖八、遠端登入主畫面

圖九顯示使用者登入系統成功畫面，因此使用者只要透電話線與數據機即可遠端登入至另一台 server，並且可遠端控制此台電腦。

```

Linux Version 2.4.8-26mdk
Compiled #1 Sat Sep 23 17:06: 17 EST 2001
One 750MHz AMD Duron Processor, 121M RAM
1507.32 Byteswap Total

adrian

Mandrake Linux release 8.1 (Villain) for i586
Kernel 2.4.8-26mdk on an i686 / i

adrian login: adrian
Password:
Last login: Fri Mar 1 19:03:27 from 140.121.114.22
adrian@adrian:~$
  
```

圖九、使用者登入系統成功

## 6. 結論與未來工作

本文描述了在 Linux 平台上製作一個具多個埠通訊伺服器之驅動程式，該程式不但提供連接各個序列埠的通訊 PPP 資料以轉向 Ethernet 經由網際網路傳送到遠端，並可將系統的設定與擷取資料經由 PHP 及 MySQL 網頁工具讓遠端控制者即時對通訊伺服器做設定；以此基礎可進一步建構出具智慧型的通訊伺服器。

## 誌 謝

茲感謝台北全濠科技股份有限公司提供相關硬體設備。

## 6. 參考文獻

- [1] Alessandro Rubini & Jonathan Corbet, “Linux Device Drivers, 2nd Edition”, O’REILLY, June 2001
- [2] Rémy Card and Frank Mével, “the LINUX KERNEL book”, WILEY, 1998
- [3] Richard Stones and Neil Matthew, “Beginning Linux Programming”, WROX, October 1999
- [4] Alessandro Rubini 原著, 歐芳吉 譯, 林長毅 編譯, “Linux 驅動程式”, O’REILLY, January 2000
- [5] Beck, Böhme, Dziadzka, Kunitz, Magnus, Verworner 原著, 張耀仁 翻譯 劉祖亮 校閱, “Linux 核心研究篇”, 和碩科技文化有限公司 出版, 學實行銷股份有限公司, December 2000
- [6] Jack Tackett, Jr. and Steven Burnett 原著, 大新資訊 譯, “Linux 超級手冊 第五冊”, 碁峰資訊股份有限公司, May 2000
- [7] 施威銘研究室 著, “Linux C 語言實務”, 旗標出版股份有限公司 出版, September 2001
- [8] 徐千洋 著, “Linux C 函式庫參考手冊”, 旗標出版股份有限公司 出版, April 2001
- [9] Andrew Sun 原著, 徐錦基 譯 蔣大偉 校編, “PPP 網路管理”, O’REILLY, January 2000
- [10] 網路邊攤 BBS 站, <telnet://openbazaar.net>

[11] 碩誠資訊 linux’s BBS 站,

<telnet://cynix.sayva.org>

[12] linuxfab, <http://www.linuxfab.cx>

[13] The Linux Kernel archives,

<http://www.Kernel.org/>